Foundation of Cryptography (0368-4162-01), Lecture 1
One-Way Functions
One-Way Functions

Iftach Haitner, Tel Aviv University

Tel Aviv University.

February 26 – March 12, 2013

Section 1

Notation

Notation I

- For $t \in \mathbb{N}$, let $[t] := \{1, \ldots, t\}$.
- Given a string $x \in \{0,1\}^*$ and $0 \leq i < j \leq |x|$, let $x_{i,\ldots,j}$ stands for the substring induced by taking the $i, \ldots, j$ bit of x (i.e., $x[i] \ldots, x[j]$).
- Given a function f defined over a set $\mathcal{U}$, and a set $\mathcal{S} \subseteq \mathcal{U}$, let $f(\mathcal{S}) := \{f(x) \colon x \in \mathcal{S}\}$, and for $y \in f(\mathcal{U})$ let $f^{-1}(y) := \{x \in \mathcal{U} \colon f(x) = y\}$.
- poly stands for the set of all polynomials.
- The worst-case running-time of a *polynomial-time algorithm* on input x, is bounded by $p(|x|)$ for some $p \in$ poly.
- A function is *polynomial-time computable*, if there exists a polynomial-time algorithm to compute it.
- PPT stands for probabilistic polynomial-time algorithms.
- A function $\mu \colon \mathbb{N} \mapsto [0,1]$ is negligible, denoted $\mu(n) = \mathrm{neg}(n)$, if for any $p \in$ poly there exists $n' \in \mathbb{N}$ with $\mu(n) \leq 1/p(n)$ for any $n > n'$.

Distribution and random variables I

- The support of a distribution P over a finite set $\mathcal{U}$, denoted $\mathrm{Supp}(P)$, is defined as $\{u \in \mathcal{U} \colon P(u) > 0\}$.
- Given a distribution P and en event E with $\Pr_P[E] > 0$, we let $(P \mid E)$ denote the conditional distribution P given E (i.e., $(P \mid E)(x) = \frac{D(x) \wedge E}{\Pr_P[E]}$).
- For $t \in \mathbb{N}$, let let U_t denote a random variable uniformly distributed over $\{0,1\}^t$.
- Given a random variable X, we let $x \leftarrow X$ denote that x is distributed according to X (e.g., $\Pr_{x \leftarrow X}[x = 7]$).
- Given a final set $\mathcal{S}$, we let $x \leftarrow \mathcal{S}$ denote that x is uniformly distributed in $\mathcal{S}$.
- We use the convention that when a random variable appears twice in the same expression, it refers to a *single* instance of this random variable. For instance, $\Pr[X = X] = 1$ (regardless of the definition of X).

Distribution and random variables II

- Given distribution P over $\mathcal{U}$ and $t \in \mathbb{N}$, we let P^t over $\mathcal{U}^t$ be defined by $D^t(x_1, \ldots, x_t) = \Pi_{i \in [t]} D(x_i)$.
- Similarly, given a random variable X, we let X^t denote the random variable induced by t independent samples from X.

Section 2

One Way Functions

One-Way Functions

Definition 1 (One-Way Functions (OWFs))

A polynomial-time computable function $f\colon \{0,1\}^* \mapsto f\colon \{0,1\}^*$ is one-way, if

$$\Pr_{x \leftarrow \{0,1\}^n}\left[\mathsf{A}(1^n, f(x)) \in f^{-1}(f(x))\right] = \mathsf{neg}(n)$$

for any PPT A.

polynomial-time computable: there exists a polynomial-time algorithm F, such that $F(x) = f(x)$ for every $x \in \{0,1\}^*$

PPT **:** probabilistic polynomial-time algorithm

neg**:** a function $\mu\colon \mathbb{N} \mapsto [0,1]$ is a *negligible* function of n, denoted $\mu(n) = \mathsf{neg}(n)$, if for any $p \in \mathsf{poly}$ there exists $n' \in \mathbb{N}$ such that $g(n) < 1/p(n)$ for all $n > n'$

* We will typically omit 1^n from the parameter list of A

1. Is this the right definition?

1. Is this the right definition?
   - Asymptotic

1. Is this the right definition?
   - Asymptotic
   - Efficiently computable

1. Is this the right definition?
    - Asymptotic
    - Efficiently computable
    - On the average

1. Is this the right definition?
   - Asymptotic
   - Efficiently computable
   - On the average
   - Only against PPT's

1. Is this the right definition?
   - Asymptotic
   - Efficiently computable
   - On the average
   - Only against PPT's
2. OWF $\implies \mathcal{P} \neq \mathcal{NP}$?

1. Is this the right definition?
   - Asymptotic
   - Efficiently computable
   - On the average
   - Only against PPT's
2. OWF $\implies \mathcal{P} \neq \mathcal{NP}$?
3. (most) Crypto implies OWFs

1. Is this the right definition?
   - Asymptotic
   - Efficiently computable
   - On the average
   - Only against PPT's
2. OWF $\implies \mathcal{P} \neq \mathcal{NP}$?
3. (most) Crypto implies OWFs
4. Do OWFs imply Crypto?

1. Is this the right definition?
   - Asymptotic
   - Efficiently computable
   - On the average
   - Only against PPT's
2. OWF $\implies \mathcal{P} \neq \mathcal{NP}$?
3. (most) Crypto implies OWFs
4. Do OWFs imply Crypto?
5. Where do we find them?

1. Is this the right definition?
    - Asymptotic
    - Efficiently computable
    - On the average
    - Only against PPT's
2. OWF $\Longrightarrow \mathcal{P} \neq \mathcal{NP}$?
3. (most) Crypto implies OWFs
4. Do OWFs imply Crypto?
5. Where do we find them?
6. Non uniform OWFs

1. Is this the right definition?
   - Asymptotic
   - Efficiently computable
   - On the average
   - Only against PPT's
2. OWF $\implies \mathcal{P} \neq \mathcal{NP}$?
3. (most) Crypto implies OWFs
4. Do OWFs imply Crypto?
5. Where do we find them?
6. Non uniform OWFs

Definition 2 (Non-uniform OWF))

A polynomial-time computable function $f : \{0,1\}^* \mapsto \{0,1\}^*$ is non-uniformly one-way, if

$$\Pr_{x \leftarrow \{0,1\}^n}\left[C_n(f(x)) \in f^{-1}(f(x))\right] = \text{neg}(n)$$

for any polynomial-size family of circuits $\{C_n\}_{n\in\mathbb{N}}$.

Length preserving functions

Definition 3 (length preserving functions)

A function $f\colon \{0,1\}^* \mapsto f\colon \{0,1\}^*$ is length preserving, if $|f(x)| = |x|$ for every $x \in \{0,1\}^*$

Length preserving functions

Definition 3 (length preserving functions)

A function $f\colon \{0,1\}^* \mapsto f\colon \{0,1\}^*$ is length preserving, if $|f(x)| = |x|$ for every $x \in \{0,1\}^*$

Theorem 4

Assume that OWFs exit, then there exist length-preserving OWFs

Length preserving functions

Definition 3 (length preserving functions)

A function $f\colon \{0,1\}^* \mapsto f\colon \{0,1\}^*$ is length preserving, if $|f(x)| = |x|$ for every $x \in \{0,1\}^*$

Theorem 4

Assume that OWFs exit, then there exist length-preserving OWFs

Proof idea: use the assumed OWF to create a length preserving one

Partial domain functions

Definition 5 (Partial domain functions)

For $m, \ell \colon \mathbb{N} \mapsto \mathbb{N}$, let $h \colon \{0,1\}^{m(n)} \mapsto \{0,1\}^{\ell(n)}$ denote a function defined over input lengths in $\{m(n)\}_{n \in \mathbb{N}}$, and maps strings of length $m(n)$ to strings of length $\ell(n)$.

Partial domain functions

Definition 5 (Partial domain functions)

For $m, \ell \colon \mathbb{N} \mapsto \mathbb{N}$, let $h \colon \{0,1\}^{m(n)} \mapsto \{0,1\}^{\ell(n)}$ denote a function defined over input lengths in $\{m(n)\}_{n \in \mathbb{N}}$, and maps strings of length $m(n)$ to strings of length $\ell(n)$.

The definition of one-wayness naturally extends to such functions.

OWFs imply Length Preserving OWFs cont.

Let $f\colon \{0,1\}^* \mapsto \{0,1\}^*$ be a OWF, let $p \in \text{poly}$ be a bound on its computing-time and assume wlg. that p is monotony increasing (can we?).

OWFs imply Length Preserving OWFs cont.

Let $f\colon \{0,1\}^* \mapsto \{0,1\}^*$ be a OWF, let $p \in \text{poly}$ be a bound on its computing-time and assume wlg. that p is monotony increasing (can we?).

Construction 6 (the length preserving function)

Define $g\colon \{0,1\}^{p(n)} \mapsto \{0,1\}^{p(n)}$ as

$$g(x) = f(x_{1,\ldots,n}), 0^{p(n)-\left|f(x_{1,\ldots,n})\right|}$$

OWFs imply Length Preserving OWFs cont.

Let $f\colon \{0,1\}^* \mapsto \{0,1\}^*$ be a OWF, let $p \in \text{poly}$ be a bound on its computing-time and assume wlg. that p is monotony increasing (can we?).

Construction 6 (the length preserving function)

Define $g\colon \{0,1\}^{p(n)} \mapsto \{0,1\}^{p(n)}$ as

$$g(x) = f(x_{1,\ldots,n}), 0^{p(n)-\left|f(x_{1,\ldots,n})\right|}$$

Note that g is well defined, length preserving and efficient (why?).

OWFs imply Length Preserving OWFs cont.

Let $f\colon \{0,1\}^* \mapsto \{0,1\}^*$ be a OWF, let $p \in \text{poly}$ be a bound on its computing-time and assume wlg. that p is monotony increasing (can we?).

Construction 6 (the length preserving function)

Define $g\colon \{0,1\}^{p(n)} \mapsto \{0,1\}^{p(n)}$ as

$$g(x) = f(x_{1,\ldots,n}), 0^{p(n)-\left|f(x_{1,\ldots,n})\right|}$$

Note that g is well defined, length preserving and efficient (why?).

Claim 7

g is one-way.

OWFs imply Length Preserving OWFs cont.

Let $f\colon \{0,1\}^* \mapsto \{0,1\}^*$ be a OWF, let $p \in \text{poly}$ be a bound on its computing-time and assume wlg. that p is monotony increasing (can we?).

Construction 6 (the length preserving function)

Define $g\colon \{0,1\}^{p(n)} \mapsto \{0,1\}^{p(n)}$ as

$$g(x) = f(x_{1,\ldots,n}), 0^{p(n)-\left|f(x_{1,\ldots,n})\right|}$$

Note that g is well defined, length preserving and efficient (why?).

Claim 7

g is one-way.

How can we prove that g is one-way?

OWFs imply Length Preserving OWFs cont.

Let $f\colon \{0,1\}^* \mapsto \{0,1\}^*$ be a OWF, let $p \in$ poly be a bound on its computing-time and assume wlg. that p is monotony increasing (can we?).

Construction 6 (the length preserving function)

Define $g\colon \{0,1\}^{p(n)} \mapsto \{0,1\}^{p(n)}$ as

$$g(x) = f(x_{1,\ldots,n}), 0^{p(n)-\left|f(x_{1,\ldots,n})\right|}$$

Note that g is well defined, length preserving and efficient (why?).

Claim 7

g is one-way.

How can we prove that g is one-way?
Answer: using reduction.

Proving that g is one-way

Proof:

Assume that g is not one-way. Namely, there exists PPT A, $q \in$ poly and infinite set $\mathcal{I} \subseteq \{p(n) \colon n \in \mathbb{N}\}$, with

$$\Pr_{x \leftarrow \{0,1\}^n}\left[\mathsf{A}(y) \in g^{-1}(g(x))\right] > 1/q(n) \tag{1}$$

for every $n \in \mathcal{I}$.

Proving that g is one-way

Proof:
Assume that g is not one-way. Namely, there exists PPT A, $q \in \text{poly}$ and infinite set $\mathcal{I} \subseteq \{p(n) \colon n \in \mathbb{N}\}$, with

$$\Pr_{x \leftarrow \{0,1\}^n}\left[\mathsf{A}(y) \in g^{-1}(g(x))\right] > 1/q(n) \tag{1}$$

for every $n \in \mathcal{I}$.

We show how to use A for inverting f.

Algorithm 8 (The inverter B)

Input: 1^n and $y \in \{0,1\}^*$

1. Let $x = \mathsf{A}(1^{p(n)}, y, 0^{p(n)-|y|})$
2. Return $x_{1,\ldots,n}$

Algorithm 8 (The inverter B)

Input: 1^n and $y \in \{0,1\}^*$

1. Let $x = \mathsf{A}(1^{p(n)}, y, 0^{p(n)-|y|})$
2. Return $x_{1,\dots,n}$

Claim 9

Let $\mathcal{I}' := \{n \in \mathbb{N} \colon p(n) \in \mathcal{I}\}$. Then

1. $\mathcal{I}'$ is infinite
2. $\Pr_{x \leftarrow \{0,1\}^n}[\mathsf{B}(1^n, f(x)) \in f^{-1}(f(x))] > 1/q(p(n))$ for every $n \in \mathcal{I}'$

Algorithm 8 (The inverter B)

Input: 1^n and $y \in \{0,1\}^*$

1. Let $x = \mathsf{A}(1^{p(n)}, y, 0^{p(n)-|y|})$
2. Return $x_{1,\ldots,n}$

Claim 9

Let $\mathcal{I}' := \{n \in \mathbb{N} \colon p(n) \in \mathcal{I}\}$. Then

1. $\mathcal{I}'$ is infinite
2. $\Pr_{x \leftarrow \{0,1\}^n}[\mathsf{B}(1^n, f(x)) \in f^{-1}(f(x))] > 1/q(p(n))$ for every $n \in \mathcal{I}'$

This contradict the assumed one-wayness of f. $\square$

Algorithm 8 (The inverter B)

Input: 1^n and $y \in \{0,1\}^*$

1. Let $x = \mathsf{A}(1^{p(n)}, y, 0^{p(n)-|y|})$
2. Return $x_{1,\ldots,n}$

Claim 9

Let $\mathcal{I}' := \{n \in \mathbb{N} \colon p(n) \in \mathcal{I}\}$. Then

1. $\mathcal{I}'$ is infinite
2. $\Pr_{x \leftarrow \{0,1\}^n}[\mathsf{B}(1^n, f(x)) \in f^{-1}(f(x))] > 1/q(p(n))$ for every $n \in \mathcal{I}'$

This contradict the assumed one-wayness of f. $\square$

Proof: (1) is clear

Algorithm 8 (The inverter B)

Input: 1^n and $y \in \{0,1\}^*$

1. Let $x = \mathsf{A}(1^{p(n)}, y, 0^{p(n)-|y|})$
2. Return $x_{1,\ldots,n}$

Claim 9

Let $\mathcal{I}' := \{n \in \mathbb{N} \colon p(n) \in \mathcal{I}\}$. Then

1. $\mathcal{I}'$ is infinite
2. $\Pr_{x \leftarrow \{0,1\}^n}[\mathsf{B}(1^n, f(x)) \in f^{-1}(f(x))] > 1/q(p(n))$ for every $n \in \mathcal{I}'$

This contradict the assumed one-wayness of f. $\square$

Proof: (1) is clear, (2)

Algorithm 8 (The inverter B)

Input: 1^n and $y \in \{0,1\}^*$

1. Let $x = A(1^{p(n)}, y, 0^{p(n)-|y|})$
2. Return $x_{1,\dots,n}$

Claim 9

Let $\mathcal{I}' := \{n \in \mathbb{N} \colon p(n) \in \mathcal{I}\}$. Then

1. $\mathcal{I}'$ is infinite
2. $\Pr_{x \leftarrow \{0,1\}^n}[B(1^n, f(x)) \in f^{-1}(f(x))] > 1/q(p(n))$ for every $n \in \mathcal{I}'$

This contradict the assumed one-wayness of f. $\square$

Proof: (1) is clear, (2)

$$\Pr_{x \leftarrow \{0,1\}^n}[B(1^n, f(x)) \in f^{-1}(f(x))]$$

$$= \Pr_{x \leftarrow \{0,1\}^n}[A(1^{p(n)}, f(x), 0^{p(n)-n})_{1,\dots,n} \in f^{-1}(f(x))]$$

Algorithm 8 (The inverter B)

Input: 1^n and $y \in \{0,1\}^*$

1. Let $x = \mathsf{A}(1^{p(n)}, y, 0^{p(n)-|y|})$
2. Return $x_{1,\ldots,n}$

Claim 9

Let $\mathcal{I}' := \{n \in \mathbb{N} \colon p(n) \in \mathcal{I}\}$. Then

1. $\mathcal{I}'$ is infinite
2. $\Pr_{x \leftarrow \{0,1\}^n}[\mathsf{B}(1^n, f(x)) \in f^{-1}(f(x))] > 1/q(p(n))$ for every $n \in \mathcal{I}'$

This contradict the assumed one-wayness of f. $\square$

Proof: (1) is clear, (2)

$$\begin{aligned} &\Pr_{x \leftarrow \{0,1\}^n}[\mathsf{B}(1^n, f(x)) \in f^{-1}(f(x))] \\ &= \Pr_{x \leftarrow \{0,1\}^n}[\mathsf{A}(1^{p(n)}, f(x), 0^{p(n)-n})_{1,\ldots,n} \in f^{-1}(f(x))] \\ &\geq \Pr_{x' \leftarrow \{0,1\}^{p(n)}}[\mathsf{A}(1^{p(n)}, g(x)) \in g^{-1}(g(x))] \end{aligned}$$

Algorithm 8 (The inverter B)

Input: 1^n and $y \in \{0,1\}^*$

1. Let $x = \mathsf{A}(1^{p(n)}, y, 0^{p(n)-|y|})$
2. Return $x_{1,\ldots,n}$

Claim 9

Let $\mathcal{I}' := \{n \in \mathbb{N} \colon p(n) \in \mathcal{I}\}$. Then

1. $\mathcal{I}'$ is infinite
2. $\Pr_{x \leftarrow \{0,1\}^n}[\mathsf{B}(1^n, f(x)) \in f^{-1}(f(x))] > 1/q(p(n))$ for every $n \in \mathcal{I}'$

This contradict the assumed one-wayness of f. $\square$

Proof: (1) is clear, (2)

$$\begin{aligned}
&\Pr_{x \leftarrow \{0,1\}^n}[\mathsf{B}(1^n, f(x)) \in f^{-1}(f(x))] \\
&= \Pr_{x \leftarrow \{0,1\}^n}[\mathsf{A}(1^{p(n)}, f(x), 0^{p(n)-n})_{1,\ldots,n} \in f^{-1}(f(x))] \\
&\geq \Pr_{x' \leftarrow \{0,1\}^{p(n)}}[\mathsf{A}(1^{p(n)}, g(x)) \in g^{-1}(g(x))] \geq 1/q(p(n))
\end{aligned}$$

Conclusion

Remark 10

- We directly related the hardness of f to that of g
- The reduction is not “security preserving"

From partial domain functions to all-length functions

Construction 11

Given a function $f\colon \{0,1\}^{\ell(n)} \mapsto \{0,1\}^{\ell(n)}$, define $f_{\mathsf{all}}\colon \{0,1\}^* \mapsto \{0,1\}^*$ as

$$f_{\mathsf{all}}(x) = f(x_{1,\ldots,k}), 0^{n-k}$$

where $n = |x|$ and $k := \max\{\ell(n') \leq n \colon n' \in [n]\}$.

From partial domain functions to all-length functions

Construction 11

Given a function $f\colon \{0,1\}^{\ell(n)} \mapsto \{0,1\}^{\ell(n)}$, define $f_{\mathrm{all}}\colon \{0,1\}^* \mapsto \{0,1\}^*$ as

$$f_{\mathrm{all}}(x) = f(x_{1,\ldots,k}), 0^{n-k}$$

where $n = |x|$ and $k := \max\{\ell(n') \leq n\colon n' \in [n]\}$.

Clearly, f_{all} is length preserving defined for every input length, and efficient (i.e., poly-time computable) in case f and ℓ are.

From partial domain functions to all-length functions

Construction 11

Given a function $f\colon \{0,1\}^{\ell(n)} \mapsto \{0,1\}^{\ell(n)}$, define $f_{\mathrm{all}}\colon \{0,1\}^* \mapsto \{0,1\}^*$ as

$$f_{\mathrm{all}}(x) = f(x_{1,\ldots,k}), 0^{n-k}$$

where $n = |x|$ and $k := \max\{\ell(n') \leq n \colon n' \in [n]\}$.

Clearly, f_{all} is length preserving defined for every input length, and efficient (i.e., poly-time computable) in case f and ℓ are.

Claim 12

Assume f and ℓ are efficiently computable, f is one-way, and ℓ satisfies $1 \leq \frac{\ell(n+1)}{\ell(n)} \leq p(n)$ for some $p \in \mathrm{poly}$, then f_{all} is one-way function.

From partial domain functions to all-length functions

Construction 11

Given a function $f\colon \{0,1\}^{\ell(n)} \mapsto \{0,1\}^{\ell(n)}$, define $f_{\mathrm{all}}\colon \{0,1\}^* \mapsto \{0,1\}^*$ as

$$f_{\mathrm{all}}(x) = f(x_{1,\ldots,k}), 0^{n-k}$$

where $n = |x|$ and $k := \max\{\ell(n') \leq n\colon n' \in [n]\}$.

Clearly, f_{all} is length preserving defined for every input length, and efficient (i.e., poly-time computable) in case f and ℓ are.

Claim 12

Assume f and ℓ are efficiently computable, f is one-way, and ℓ satisfies $1 \leq \frac{\ell(n+1)}{\ell(n)} \leq p(n)$ for some $p \in \mathrm{poly}$, then f_{all} is one-way function.

Proof: ?

Weak One Way Functions

Definition 13 (weak one-way functions)

A poly-time computable function $f\colon \{0,1\}^* \mapsto f\colon \{0,1\}^*$ is α-one-way, if

$$\Pr_{x \leftarrow \{0,1\}^n}\left[\mathsf{A}(1^n, f(x)) \in f^{-1}(f(x))\right] \leq \alpha(n)$$

for any PPT A and large enough $n \in \mathbb{N}$.

Weak One Way Functions

Definition 13 (weak one-way functions)

A poly-time computable function $f\colon \{0,1\}^* \mapsto f\colon \{0,1\}^*$ is α-one-way, if

$$\Pr_{x \leftarrow \{0,1\}^n}\left[\mathsf{A}(1^n, f(x)) \in f^{-1}(f(x))\right] \leq \alpha(n)$$

for any PPT A and large enough $n \in \mathbb{N}$.

1. (strong) OWF according to Definition 1, are neg(n)-one-way according to the above definition

Weak One Way Functions

Definition 13 (weak one-way functions)

A poly-time computable function $f\colon \{0,1\}^* \mapsto f\colon \{0,1\}^*$ is α-one-way, if

$$\Pr_{x \leftarrow \{0,1\}^n}\left[A(1^n, f(x)) \in f^{-1}(f(x))\right] \leq \alpha(n)$$

for any PPT A and large enough $n \in \mathbb{N}$.

1. (strong) OWF according to Definition 1, are neg(n)-one-way according to the above definition
2. Can we “amplify" weak OWF to strong ones?

Strong to weak OWFs

Claim 14

Assume there exists OWFs, then there exist functions that are $\frac{2}{3}$-one-way, but not (strong) one-way

Strong to weak OWFs

Claim 14

Assume there exists OWFs, then there exist functions that are $\frac{2}{3}$-one-way, but not (strong) one-way

Proof: For a OWF f, let

$$g(x) = \begin{cases} (1, f(x)), & x_1 = 1; \\ 0, & \text{otherwise.} \end{cases}$$

Weak to Strong OWFs

Theorem 15

Assume there exists $(1-\alpha)$-weak OWFs with $\alpha(n) > 1/p(n)$ for some $p \in \text{poly}$, then there exists (strong) one-way functions.

Weak to Strong OWFs

Theorem 15

Assume there exists $(1-\alpha)$-weak OWFs with $\alpha(n) > 1/p(n)$ for some $p \in \mathrm{poly}$, then there exists (strong) one-way functions.

Proof: we assume wlg that f is length preserving (why can we do so?)

Weak to Strong OWFs

Theorem 15

Assume there exists $(1-\alpha)$-weak OWFs with $\alpha(n) > 1/p(n)$ for some $p \in \mathrm{poly}$, then there exists (strong) one-way functions.

Proof: we assume wlg that f is length preserving (why can we do so?)

Construction 16 (g – the strong one-way function)

Let $t \colon \mathbb{N} \mapsto \mathbb{N}$ be a poly-time computable function satisfying $t(n) \in \omega(\log n/\alpha(n))$. Define $g \colon (\{0,1\}^n)^{t(n)} \mapsto (\{0,1\}^n)^{t(n)}$ as

$$g(x_1,\ldots,x_t) = f(x_1),\ldots,f(x_t)$$

Weak to Strong OWFs

Theorem 15

Assume there exists $(1-\alpha)$-weak OWFs with $\alpha(n) > 1/p(n)$ for some $p \in \text{poly}$, then there exists (strong) one-way functions.

Proof: we assume wlg that f is length preserving (why can we do so?)

Construction 16 (g – the strong one-way function)

Let $t\colon \mathbb{N} \mapsto \mathbb{N}$ be a poly-time computable function satisfying $t(n) \in \omega(\log n/\alpha(n))$. Define $g\colon (\{0,1\}^n)^{t(n)} \mapsto (\{0,1\}^n)^{t(n)}$ as

$$g(x_1, \ldots, x_t) = f(x_1), \ldots, f(x_t)$$

Claim 17

g is one-way.

Proving that g is one-way – the naive approach

Let A be a potential inverter for g, and assume that A tries to attacks each of the t outputs of g independently. Then

$$\Pr_{x \leftarrow \{0,1\}^{t(n)\cdot n}}[\mathsf{A}(g(x)) \in g^{-1}(g(x))] \leq (1-\alpha(n))^{t(n)} \leq e^{-\omega(\log n)} = \operatorname{neg}(n)$$

Proving that g is one-way – the naive approach

Let A be a potential inverter for g, and assume that A tries to attacks each of the t outputs of g independently. Then

$$\Pr_{x \leftarrow \{0,1\}^{t(n)\cdot n}}[\mathsf{A}(g(x)) \in g^{-1}(g(x))] \leq (1-\alpha(n))^{t(n)} \leq e^{-\omega(\log n)} = \operatorname{neg}(n)$$

A less naive approach would be to assume that A goes over output sequentially.

Proving that g is one-way – the naive approach

Let A be a potential inverter for g, and assume that A tries to attacks each of the t outputs of g independently. Then

$$\Pr_{x \leftarrow \{0,1\}^{t(n)\cdot n}}[A(g(x)) \in g^{-1}(g(x))] \leq (1-\alpha(n))^{t(n)} \leq e^{-\omega(\log n)} = \mathrm{neg}(n)$$

A less naive approach would be to assume that A goes over output sequentially.

Unfortunately, we can assume none of the above.

Proving that g is one-way – the naive approach

Let A be a potential inverter for g, and assume that A tries to attacks each of the t outputs of g independently. Then

$$\Pr_{x \leftarrow \{0,1\}^{t(n)\cdot n}}[\mathsf{A}(g(x)) \in g^{-1}(g(x))] \leq (1-\alpha(n))^{t(n)} \leq e^{-\omega(\log n)} = \mathrm{neg}(n)$$

A less naive approach would be to assume that A goes over output sequentially.

Unfortunately, we can assume none of the above.

Any idea?

Failing Sets

Failing Sets

Definition 18 (failing set)

A function $f\colon \{0,1\}^n \mapsto \{0,1\}^{\ell(n)}$ has a (δ, ε)-failing set for algorithm A, if for large enough n, exists set $\mathcal{S} = \mathcal{S}(n) \subseteq \{0,1\}^{\ell(n)}$ with

1. $\Pr_{x \leftarrow \{0,1\}^n}[f(x) \in \mathcal{S}] \geq \delta(n)$, and
2. $\Pr[\mathsf{A}(y) \in f^{-1}(y)] < \varepsilon(n)$, for every $y \in \mathcal{S}$

Failing Sets

Definition 18 (failing set)

A function $f\colon \{0,1\}^n \mapsto \{0,1\}^{\ell(n)}$ has a (δ,ε)-failing set for algorithm A, if for large enough n, exists set $\mathcal{S} = \mathcal{S}(n) \subseteq \{0,1\}^{\ell(n)}$ with

1. $\Pr_{x \leftarrow \{0,1\}^n}[f(x) \in \mathcal{S}] \geq \delta(n)$, and
2. $\Pr[\mathsf{A}(y) \in f^{-1}(y)] < \varepsilon(n)$, for every $y \in \mathcal{S}$

Claim 19

Let f be a $(1-\alpha)$-OWF. Then f has $(\alpha/2, 1/p)$-failing set for any PPT A and $p \in \mathrm{poly}$.

Failing Sets

Definition 18 (failing set)

A function $f\colon \{0,1\}^n \mapsto \{0,1\}^{\ell(n)}$ has a (δ, ε)-failing set for algorithm A, if for large enough n, exists set $\mathcal{S} = \mathcal{S}(n) \subseteq \{0,1\}^{\ell(n)}$ with

1. $\Pr_{x \leftarrow \{0,1\}^n}[f(x) \in \mathcal{S}] \geq \delta(n)$, and
2. $\Pr[\mathsf{A}(y) \in f^{-1}(y)] < \varepsilon(n)$, for every $y \in \mathcal{S}$

Claim 19

Let f be a $(1-\alpha)$-OWF. Then f has $(\alpha/2, 1/p)$-failing set for any PPT A and $p \in \mathrm{poly}$.

Proof: Assume $\exists$ PPT A, $p \in \mathrm{poly}$ and an infinite set $\mathcal{I} \subseteq \mathbb{N}$ such that for every $n \in \mathcal{I}$, $\exists \mathcal{L} \subseteq \{0,1\}^n$ with

1. $\Pr_{x \leftarrow \{0,1\}^n}[f(x) \in \mathcal{L}] \geq 1 - \alpha(n)/2$, and
2. $\Pr[\mathsf{A}(y) \in f^{-1}(y)] \geq 1/p(n)$, for every $y \in \mathcal{L}$

Failing Sets

Definition 18 (failing set)

A function $f\colon \{0,1\}^n \mapsto \{0,1\}^{\ell(n)}$ has a (δ,ε)-failing set for algorithm A, if for large enough n, exists set $\mathcal{S} = \mathcal{S}(n) \subseteq \{0,1\}^{\ell(n)}$ with

1. $\Pr_{x \leftarrow \{0,1\}^n}[f(x) \in \mathcal{S}] \geq \delta(n)$, and
2. $\Pr[\mathsf{A}(y) \in f^{-1}(y)] < \varepsilon(n)$, for every $y \in \mathcal{S}$

Claim 19

Let f be a $(1-\alpha)$-OWF. Then f has $(\alpha/2, 1/p)$-failing set for any PPT A and $p \in \text{poly}$.

Proof: Assume $\exists$ PPT A, $p \in \text{poly}$ and an infinite set $\mathcal{I} \subseteq \mathbb{N}$ such that for every $n \in \mathcal{I}$, $\exists \mathcal{L} \subseteq \{0,1\}^n$ with

1. $\Pr_{x \leftarrow \{0,1\}^n}[f(x) \in \mathcal{L}] \geq 1 - \alpha(n)/2$, and
2. $\Pr[\mathsf{A}(y) \in f^{-1}(y)] \geq 1/p(n)$, for every $y \in \mathcal{L}$

We'll use A to contradict the hardness of f.

Using A to invert f

Using A to invert f

Algorithm 20 (The inverter B)

Input: $y \in \{0,1\}^n$.
Do (with fresh randomness) for $n \cdot p(n)$ times:
If $x = A(y) \in f^{-1}(y)$, return x

Using A to invert f

Algorithm 20 (The inverter B)

Input: $y \in \{0,1\}^n$.
Do (with fresh randomness) for $n \cdot p(n)$ times:
If $x = \mathsf{A}(y) \in f^{-1}(y)$, return x

Clearly, B is a PPT

Using A to invert f

Algorithm 20 (The inverter B)

Input: $y \in \{0,1\}^n$.
Do (with fresh randomness) for $n \cdot p(n)$ times:
If $x = \mathsf{A}(y) \in f^{-1}(y)$, return x

Clearly, B is a PPT

Claim 21

For every large enough $n \in \mathcal{I}$, it holds that
$\Pr_{x \leftarrow \{0,1\}^n}\left[\mathsf{B}(f(x)) \in f^{-1}(f(x))\right] > 1 - \alpha(n)$

Using A to invert f

Algorithm 20 (The inverter B)

Input: $y \in \{0,1\}^n$.
Do (with fresh randomness) for $n \cdot p(n)$ times:
If $x = A(y) \in f^{-1}(y)$, return x

Clearly, B is a PPT

Claim 21

For every large enough $n \in \mathcal{I}$, it holds that
$\Pr_{x \leftarrow \{0,1\}^n}\left[B(f(x)) \in f^{-1}(f(x))\right] > 1 - \alpha(n)$

Hence, f is not $(1-\alpha)$-one-way $\square$

Proof: [of Claim 21]
All probabilities below are also over $y \leftarrow f(x); x \leftarrow \{0,1\}^n$:

Proof: [of Claim 21]
All probabilities below are also over $y \leftarrow f(x);\ x \leftarrow \{0,1\}^n$:

$$\Pr[\mathsf{B}(y) \in f^{-1}(y)]$$

Proof: [of Claim 21]
All probabilities below are also over $y \leftarrow f(x)$; $x \leftarrow \{0,1\}^n$:

$$\begin{aligned}
&\Pr[\mathsf{B}(y) \in f^{-1}(y)] \\
&\quad \geq \quad \Pr[\mathsf{B}(y) \in f^{-1}(y) \wedge y \in \mathcal{L}(n)]
\end{aligned}$$

Proof: [of Claim 21]
All probabilities below are also over $y \leftarrow f(x)$; $x \leftarrow \{0,1\}^n$:

$$\begin{aligned}
&\Pr[\mathsf{B}(y) \in f^{-1}(y)] \\
&\quad \geq \quad \Pr[\mathsf{B}(y) \in f^{-1}(y) \wedge y \in \mathcal{L}(n)] \\
&\quad = \quad \Pr[y \in \mathcal{L}(n)] \cdot \Pr[\mathsf{B}(y) \in f^{-1}(y) \mid y \in \mathcal{L}(n)]
\end{aligned}$$

Proof: [of Claim 21]
All probabilities below are also over $y \leftarrow f(x)$; $x \leftarrow \{0,1\}^n$:

$$
\begin{aligned}
&\Pr[\mathsf{B}(y) \in f^{-1}(y)] \\
&\quad \geq \Pr[\mathsf{B}(y) \in f^{-1}(y) \wedge y \in \mathcal{L}(n)] \\
&\quad = \Pr[y \in \mathcal{L}(n)] \cdot \Pr[\mathsf{B}(y) \in f^{-1}(y) \mid y \in \mathcal{L}(n)] \\
&\quad \geq (1 - \alpha(n)/2) \cdot (1 - (1 - 1/p(n))^{np(n)})
\end{aligned}
$$

Proof: [of Claim 21]
All probabilities below are also over $y \leftarrow f(x)$; $x \leftarrow \{0,1\}^n$:

$$\begin{aligned}
&\Pr[\mathsf{B}(y) \in f^{-1}(y)] \\
&\quad \geq \Pr[\mathsf{B}(y) \in f^{-1}(y) \wedge y \in \mathcal{L}(n)] \\
&\quad = \Pr[y \in \mathcal{L}(n)] \cdot \Pr[\mathsf{B}(y) \in f^{-1}(y) \mid y \in \mathcal{L}(n)] \\
&\quad \geq (1 - \alpha(n)/2) \cdot (1 - (1 - 1/p(n))^{np(n)}) \\
&\quad \geq (1 - \alpha(n)/2) \cdot (1 - 2^{-n}) > 1 - \alpha(n),
\end{aligned}$$

for large enough n. ♣

Proving that g is one-way

We show that if g is not OWF, then f has no flailing-set of the “right" type.

Proving that g is one-way

We show that if g is not OWF, then f has no flailing-set of the “right" type.

Claim 22

Assume $\exists$ PPT A, $p \in$ poly and an infinite set $\mathcal{I} \subseteq \mathbb{N}$ s.t.

$$\Pr_{w \leftarrow \{0,1\}^{t(n)\cdot n}}[\mathsf{A}(g(x)) \in g^{-1}(g(w))] \geq 1/p(n) \tag{2}$$

for every $n \in \mathcal{I}$.

Proving that g is one-way

We show that if g is not OWF, then f has no flailing-set of the “right" type.

Claim 22

Assume $\exists$ PPT A, $p \in$ poly and an infinite set $\mathcal{I} \subseteq \mathbb{N}$ s.t.

$$\Pr_{w \leftarrow \{0,1\}^{t(n) \cdot n}}[\mathsf{A}(g(x)) \in g^{-1}(g(w))] \geq 1/p(n) \tag{2}$$

for every $n \in \mathcal{I}$. Then $\exists$ PPT B and $q \in$ poly s.t.

$$\Pr_{y \leftarrow \mathcal{S}}[\mathsf{B}(y) \in f^{-1}(y)] \geq 1/q(n) \tag{3}$$

for every $n \in \mathcal{I}$ and $\mathcal{S} \subseteq \{0,1\}^n$ with $\Pr_{x \leftarrow \{0,1\}^n}[f(x) \in \mathcal{S}] \geq \alpha(n)/2$.

Proving that g is one-way

We show that if g is not OWF, then f has no flailing-set of the "right" type.

Claim 22

Assume $\exists$ PPT A, $p \in$ poly and an infinite set $\mathcal{I} \subseteq \mathbb{N}$ s.t.

$$\Pr_{w \leftarrow \{0,1\}^{t(n) \cdot n}}[\mathsf{A}(g(x)) \in g^{-1}(g(w))] \geq 1/p(n) \tag{2}$$

for every $n \in \mathcal{I}$. Then $\exists$ PPT B and $q \in$ poly s.t.

$$\Pr_{y \leftarrow \mathcal{S}}[\mathsf{B}(y) \in f^{-1}(y)] \geq 1/q(n) \tag{3}$$

for every $n \in \mathcal{I}$ and $\mathcal{S} \subseteq \{0,1\}^n$ with $\Pr_{x \leftarrow \{0,1\}^n}[f(x) \in \mathcal{S}] \geq \alpha(n)/2$.

Namely, f does not have a $(\alpha/2, 1/q)$-failing set.

Algorithm B

Algorithm 23 (No failing-set algorithm B)

Input: $y \in \{0,1\}^n$.

1. Choose $w \leftarrow \{0,1\}^{t(n)\cdot n}$, $z = (z_1,\ldots,z_t) = g(w)$ and $i \leftarrow [t]$
2. Set $z' = (z_1,\ldots,z_{i-1},y,z_{i+1},\ldots,z_t)$
3. Return $\mathsf{A}(z')_i$

Algorithm B

Algorithm 23 (No failing-set algorithm B)

Input: $y \in \{0,1\}^n$.

1. Choose $w \leftarrow \{0,1\}^{t(n)\cdot n}$, $z = (z_1, \ldots, z_t) = g(w)$ and $i \leftarrow [t]$
2. Set $z' = (z_1, \ldots, z_{i-1}, y, z_{i+1}, \ldots, z_t)$
3. Return $\mathsf{A}(z')_i$

Fix $n \in \mathcal{I}$ and a set $\mathcal{S} \subseteq \{0,1\}^n$ with $\Pr_{x \leftarrow \{0,1\}^n}[f(x) \in \mathcal{S}] \geq \alpha(n)/2$.
We analyze B's success probability with respect to $\mathcal{S}$, using the following (inefficient) algorithm B^*:

Algorithm B*

Definition 24 (Bad)

For $z = (z_1, \ldots, z_t) \in Im(g)$ (the image of g), we set $Bad(z) = 1$ iff $\nexists i \in [t]$ with $z_i \in \mathcal{S}$.

Algorithm B*

Definition 24 (Bad)

For $z = (z_1, \ldots, z_t) \in Im(g)$ (the image of g), we set $Bad(z) = 1$ iff $\nexists i \in [t]$ with $z_i \in \mathcal{S}$.

B* differ from B in the way it chooses z': in case $\mathrm{Bad}(z) = 1$, it sets $z' = z$. Otherwise, it sets i to the first index $j \in [t]$ with $z_j \in \mathcal{S}$, and sets z' as B does with respect to this i.

Algorithm B^*

Definition 24 (Bad)

For $z = (z_1, \ldots, z_t) \in Im(g)$ (the image of g), we set $Bad(z) = 1$ iff $\nexists i \in [t]$ with $z_i \in \mathcal{S}$.

B^* differ from B in the way it chooses z': in case $\mathrm{Bad}(z) = 1$, it sets $z' = z$. Otherwise, it sets i to the first index $j \in [t]$ with $z_j \in \mathcal{S}$, and sets z' as B does with respect to this i.

Claim 25

$\Pr_{y \leftarrow \mathcal{S}}[B^*(y) \in f^{-1}(y)] \geq \frac{1}{p(n)} - \mathrm{neg}(n)$,

Algorithm B^*

Definition 24 (Bad)

For $z = (z_1, \ldots, z_t) \in Im(g)$ (the image of g), we set $Bad(z) = 1$ iff $\nexists i \in [t]$ with $z_i \in \mathcal{S}$.

B^* differ from B in the way it chooses z': in case $\mathrm{Bad}(z) = 1$, it sets $z' = z$. Otherwise, it sets i to the first index $j \in [t]$ with $z_j \in \mathcal{S}$, and sets z' as B does with respect to this i.

Claim 25

$\Pr_{y \leftarrow \mathcal{S}}[B^*(y) \in f^{-1}(y)] \geq \frac{1}{p(n)} - \mathrm{neg}(n)$,

Therefore, $\Pr_{y \leftarrow \mathcal{S}}[B(y) \in f^{-1}(y)] \geq \frac{1}{t(n)p(n)} - \mathrm{neg}(n)$.$\square$

Claim 25 follows from the following two claims,

Claim 26

$\Pr_{w \leftarrow \{0,1\}^{t(n) \cdot n}}[\mathrm{Bad}(g(w))] = \mathrm{neg}(n)$

Claim 25 follows from the following two claims,

Claim 26

$\Pr_{w \leftarrow \{0,1\}^{t(n)\cdot n}}[\mathrm{Bad}(g(w))] = \mathrm{neg}(n)$

Claim 27

- Let $Z = g(W)$ for $W \leftarrow \{0,1\}^{t(n)\cdot n}$

Claim 25 follows from the following two claims,

Claim 26

$\Pr_{w \leftarrow \{0,1\}^{t(n)\cdot n}}[\mathrm{Bad}(g(w))] = \mathrm{neg}(n)$

Claim 27

- Let $Z = g(W)$ for $W \leftarrow \{0,1\}^{t(n)\cdot n}$
- Let Z' be the value of z' induced by a random execution of $\mathsf{B}^*(f(X))$, for $X \leftarrow \{0,1\}^n \mid f(X) \in \mathcal{S}$.

Claim 25 follows from the following two claims,

Claim 26

$\Pr_{w \leftarrow \{0,1\}^{t(n)\cdot n}}[\mathrm{Bad}(g(w))] = \mathrm{neg}(n)$

Claim 27

- Let $Z = g(W)$ for $W \leftarrow \{0,1\}^{t(n)\cdot n}$
- Let Z' be the value of z' induced by a random execution of $\mathsf{B}^*(f(X))$, for $X \leftarrow \{0,1\}^n \mid f(X) \in \mathcal{S}$.

Claim 25 follows from the following two claims,

Claim 26

$\Pr_{w \leftarrow \{0,1\}^{t(n)\cdot n}}[\mathrm{Bad}(g(w))] = \mathrm{neg}(n)$

Claim 27

- Let $Z = g(W)$ for $W \leftarrow \{0,1\}^{t(n)\cdot n}$
- Let Z' be the value of z' induced by a random execution of $\mathsf{B}^*(f(X)$, for $X \leftarrow \{0,1\}^n \mid f(X) \in \mathcal{S}$.

Then Z and Z' are identically distributed.

The above claims imply Claim 25.

The above claims imply Claim 25.

$$\Pr_{y \leftarrow \mathcal{S}}[\mathsf{B}^*(y) \in f^{-1}(y)] \geq \Pr_{w \leftarrow \{0,1\}^{t(n)\cdot n}; z = g(w)} \left[\mathsf{A}(z) \in g^{-1}(z) \wedge \neg\, \mathsf{Bad}(z)\right] \quad (4)$$

The above claims imply Claim 25.

$$\Pr_{y \leftarrow \mathcal{S}}[\mathsf{B}^*(y) \in f^{-1}(y)] \geq \Pr_{w \leftarrow \{0,1\}^{t(n)\cdot n}; z = g(w)}\left[\mathsf{A}(z) \in g^{-1}(z) \wedge \neg \mathsf{Bad}(z)\right] \tag{4}$$

$$\Pr_{w \leftarrow \{0,1\}^{t(n)\cdot n}; z = g(w)}\left[\mathsf{A}(z) \in g^{-1}(z)\right] \tag{5}$$

$$\leq \Pr[\mathsf{A}(z) \in g^{-1}(z) \wedge \neg \mathsf{Bad}(z)] + \Pr[\mathsf{Bad}(z)]$$

The above claims imply Claim 25.

$$\Pr_{y\leftarrow\mathcal{S}}[\mathsf{B}^*(y)\in f^{-1}(y)] \geq \Pr_{w\leftarrow\{0,1\}^{t(n)\cdot n};z=g(w)}\left[\mathsf{A}(z)\in g^{-1}(z)\wedge\neg\,\mathsf{Bad}(z)\right] \tag{4}$$

$$\begin{aligned}&\Pr_{w\leftarrow\{0,1\}^{t(n)\cdot n};z=g(w)}\left[\mathsf{A}(z)\in g^{-1}(z)\right] \\ &\leq \Pr[\mathsf{A}(z)\in g^{-1}(z)\wedge\neg\,\mathsf{Bad}(z)]+\Pr[\mathsf{Bad}(z)]\end{aligned} \tag{5}$$

It follows that

$$\begin{aligned}\Pr_{y\leftarrow\mathcal{S}}[\mathsf{B}^*(y)\in f^{-1}(y)] &\geq \Pr_{w\leftarrow\{0,1\}^{t(n)\cdot n};z=g(w)}[\mathsf{A}(z)\in g^{-1}(z)]-\operatorname{neg}(n)\\ &\geq \frac{1}{p(n)}-\operatorname{neg}(n).\square\end{aligned}$$

Proof of Claim 26?

Proof of Claim 26?

Proof of Claim 27: Let $\beta = \Pr_{x \leftarrow \{0,1\}^n}[f(x) \in \mathcal{S}]$ and consider the following awkward method to sample according to Z

Proof of Claim 26?

Proof of Claim 27: Let $\beta = \Pr_{x \leftarrow \{0,1\}^n}[f(x) \in \mathcal{S}]$ and consider the following awkward method to sample according to Z

Algorithm 28 (P)

1. Sample $\ell_1, \ldots, \ell_{t(n)}$, each taking the value 1 with β.
2. Output $z_1, \ldots, z_{t(n)}$, where z_i is sampled according to

$$\begin{cases} f(x) \mid x \leftarrow \{0,1\}^n, f(x) \in \mathcal{S}, & \ell_i = 1; \\ f(x) \mid x \leftarrow \{0,1\}^n, f(x) \notin \mathcal{S}, & \text{otherwise.} \end{cases}$$

Proof of Claim 26?

Proof of Claim 27: Let $\beta = \Pr_{x \leftarrow \{0,1\}^n}[f(x) \in \mathcal{S}]$ and consider the following awkward method to sample according to Z

Algorithm 28 (P)

1. Sample $\ell_1, \ldots, \ell_{t(n)}$, each taking the value 1 with β.
2. Output $z_1, \ldots, z_{t(n)}$, where z_i is sampled according to

$$\begin{cases} f(x) \mid x \leftarrow \{0,1\}^n, f(x) \in \mathcal{S}, & \ell_i = 1; \\ f(x) \mid x \leftarrow \{0,1\}^n, f(x) \notin \mathcal{S}, & \text{otherwise.} \end{cases}$$

The process for sampling Z' can be described as follows:

1. Choose $\ell_1, \ldots, \ell_{t(n)}$ and $z_1, \ldots, z_{t(n)}$ according to P
2. Resample z_i for some i with $\ell_i = 1$ (if such exists)

Proof of Claim 26?

Proof of Claim 27: Let $\beta = \Pr_{x \leftarrow \{0,1\}^n}[f(x) \in \mathcal{S}]$ and consider the following awkward method to sample according to Z

Algorithm 28 (P)

1. Sample $\ell_1, \ldots, \ell_{t(n)}$, each taking the value 1 with β.
2. Output $z_1, \ldots, z_{t(n)}$, where z_i is sampled according to

$$\begin{cases} f(x) \mid x \leftarrow \{0,1\}^n, f(x) \in \mathcal{S}, & \ell_i = 1; \\ f(x) \mid x \leftarrow \{0,1\}^n, f(x) \notin \mathcal{S}, & \text{otherwise.} \end{cases}$$

The process for sampling Z' can be described as follows:

1. Choose $\ell_1, \ldots, \ell_{t(n)}$ and $z_1, \ldots, z_{t(n)}$ according to P
2. Resample z_i for some i with $\ell_i = 1$ (if such exists)

Hence, Z' has the same distribution as of P, and hence as of Z. $\square$

Conclusion

Remark 29 (hardness amplification via parallel repetition)

- Can we give a more efficient (secure) reduction?

Conclusion

Remark 29 (hardness amplification via parallel repetition)

- Can we give a more efficient (secure) reduction?
- Similar theorems for other cryptographic primitives (e.g., Captchas, general protocols)?

Conclusion

Remark 29 (hardness amplification via parallel repetition)

- Can we give a more efficient (secure) reduction?
- Similar theorems for other cryptographic primitives (e.g., Captchas, general protocols)?
  What properties of the weak OWF have we used in the proof?